Make Projects
build, hack, tweak, share, discover.

3D Models with OpenSCAD

Written By: Sean Michael Ragan

TOOLS:

- Computer (1)
- Drawing software (1)
- OpenSCAD (1)

SUMMARY

I wrote recently about the excitement surrounding Clifford Wolf's new freeware OpenSCAD program. OpenSCAD uses a cool keep-it-super-simple approach to 3D modeling, eliminating the resource-hungry what-you-see-is-what-you-get (WYSIWYG) editing environment favored by most 3D modeling packages, and replacing it with a text-based scripting environment in which models are programmed, instead of sculpted. Basically, you write a script describing your model's shape and then compile it to produce the actual model, which is then rendered onscreen and can be exported to STL format for 3D printing or other purposes.

OpenSCAD has two powerful features to facilitate this programming process. The first is support for so-called "constructive solid geometry" (CSG) modeling, in which complex forms are built up as intersections, unions, and differences of simple primary shapes like boxes, cylinders, cones, and ellipsoids. If you've ever used the ray-tracing program POV-Ray before, this idea will be familiar to you.

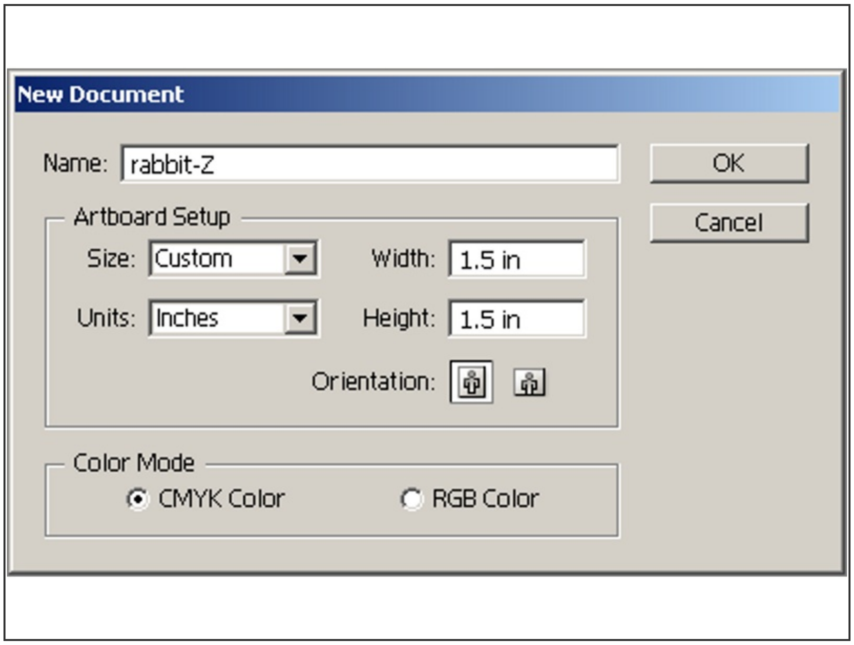
The second, less-well-publicized (but perhaps equally powerful) feature of OpenSCAD is

"DXF extrusion," in which OpenSCAD will import a 2D drawing in AutoCAD's popular drawing exchange format (DXF) and "extrude" it into the third dimension. OpenSCAD has support for linear extrusion, in which the resulting part has straight vertical sides, and also rotating extrusion, which results in a part with helical sides. Since a large number of models for rapid prototyping are simple extruded profiles, I expect this feature to see a lot of use.

In this tutorial, I'm going to show you how to use OpenSCAD to produce a simple 3D model by extruding a part profile produced in normal drawing software. I use Adobe Illustrator CS3 because I have access to it and am familiar with its interface, but the popular freeware drawing program InkScape will read and write DXF files natively, and there's no reason why it couldn't serve just as well if you prefer it. There are a number of other free and low-cost programs that will export DXF files. OpenSCAD's developer mentions QCAD, which, as of this writing, is available from its developer RibbonSoft for €24.

The part I'm making is one of 12 solid pentomino puzzle pieces based on the animals of the Chinese zodiac--in this case, the rabbit or "Z" pentomino. The designs are based on those of Japanese schoolteacher Sabu Oguro as published on p. 40 of Jerry Slocum and Jack Botermans' 1986 book *Puzzles Old & New: How to Make and Solve Them*, an image from which is reproduced at the top of this article. My original DXF files and the extruded 3D STL files are freely available for download at Thingiverse. Becky Stern printed all the real-world models shown in this article on her MakerBot Cupcake 3D printer, and photographed them for this tutorial.

Step 1 — Size art board appropriately



Name: rabbit-Z

Size: Custom

Width: 1.5 in

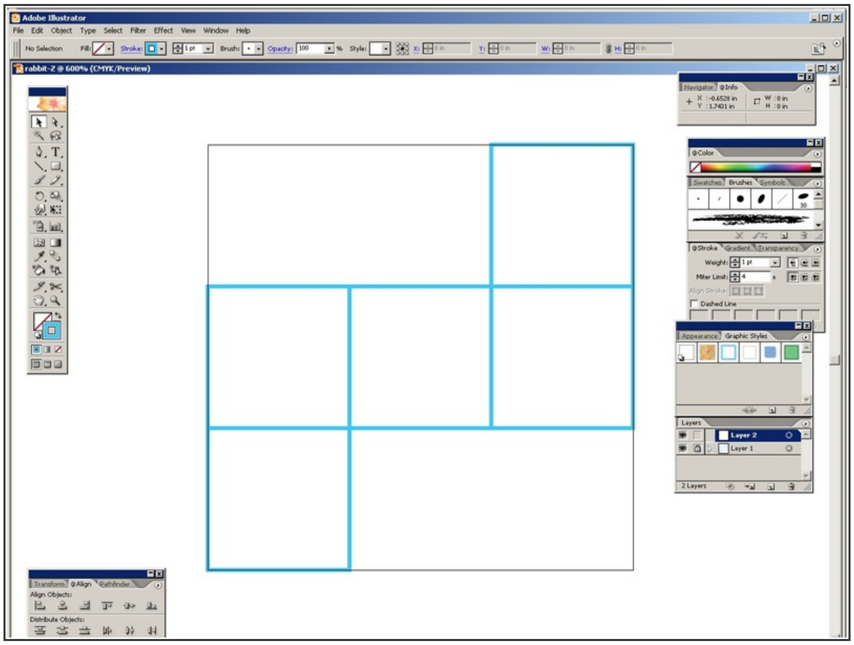
Units: Inches

Height: 1.5 in

● CMYK Color ○ RGB Color

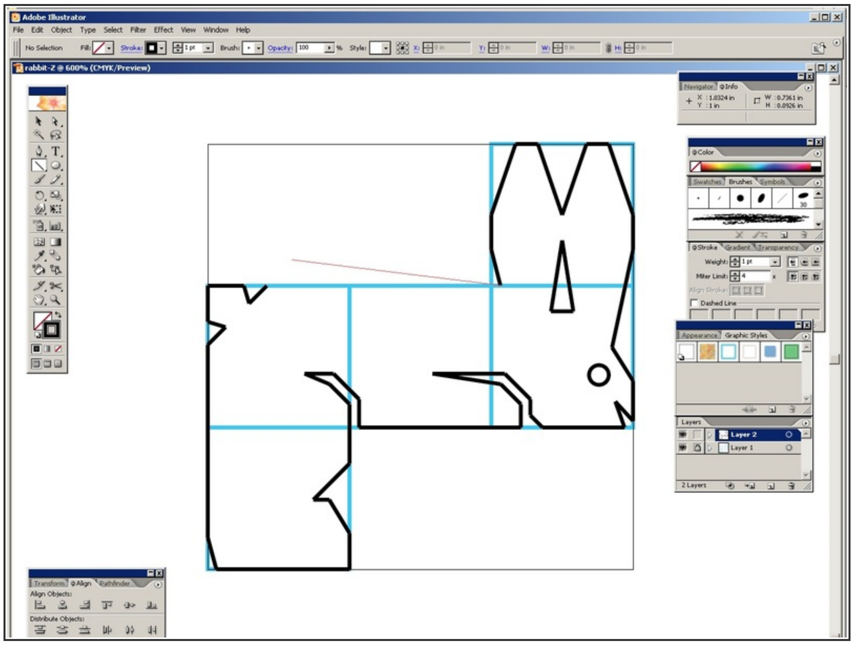
- Sizing the page or art board to exactly the size of your finished part will make it easier to view the rendered extrusion in OpenSCAD. The zero coordinate seems to be determined based on one corner of the art board, and if you draw a 1.5" x 1.5" part in the middle of a letter-sized page, the actual model tends to end up out off the edge of the screen when you render it. Save yourself the nuisance and shrink the art board down when you first create the new drawing.

Step 2 — Draw grid in bottom layer



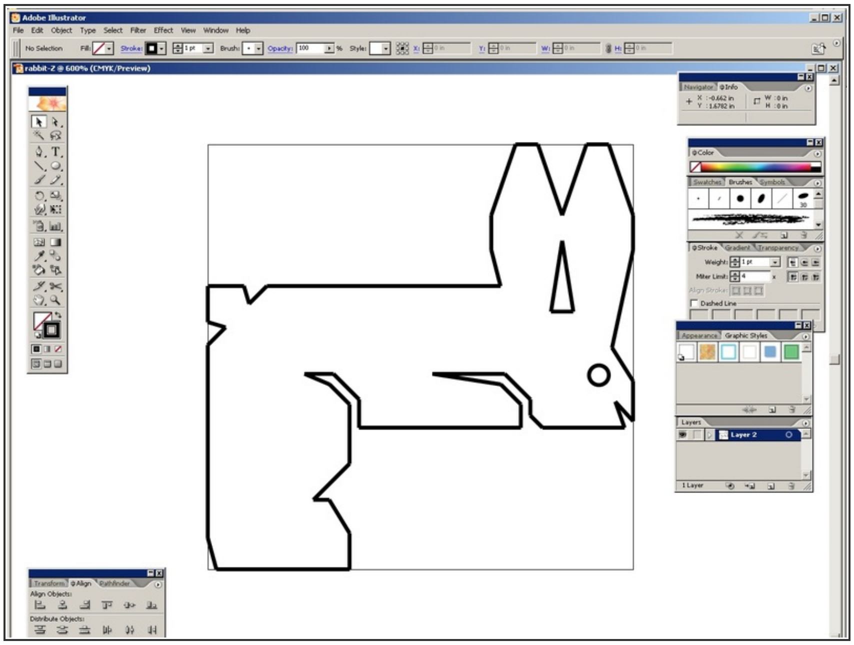
- A grid or any other drawing guides you may want to include should be drawn first, in a separate layer.
- In the case of the rabbit pentomino, the animal profile needs to fit into the outline of the normal "Z" pentomino, which is five squares simply joined as shown.
- These puzzle pieces are designed on a 0.5" grid, so start by drawing a single 0.5" square using the box tool. Then make a copy of that square, paste it, and use the arrow tool to snap its corners to the corners of the original square. Then repeat the copy-paste-snap process to add the remaining three squares in their proper positions. Then "select all" and change the stroke color to something other than black to make it easy to distinguish part lines from grid lines later on.
- Finally, lock the layer and create a new one to draw the actual part lines in. Change the stroke color back to black.

Step 3 — Draw profile without using splines or polylines


Adobe Illustrator
File Edit Object Type Select Filter Effect View Window Help
rabbit-Z @ 600% (CMYK/Preview)

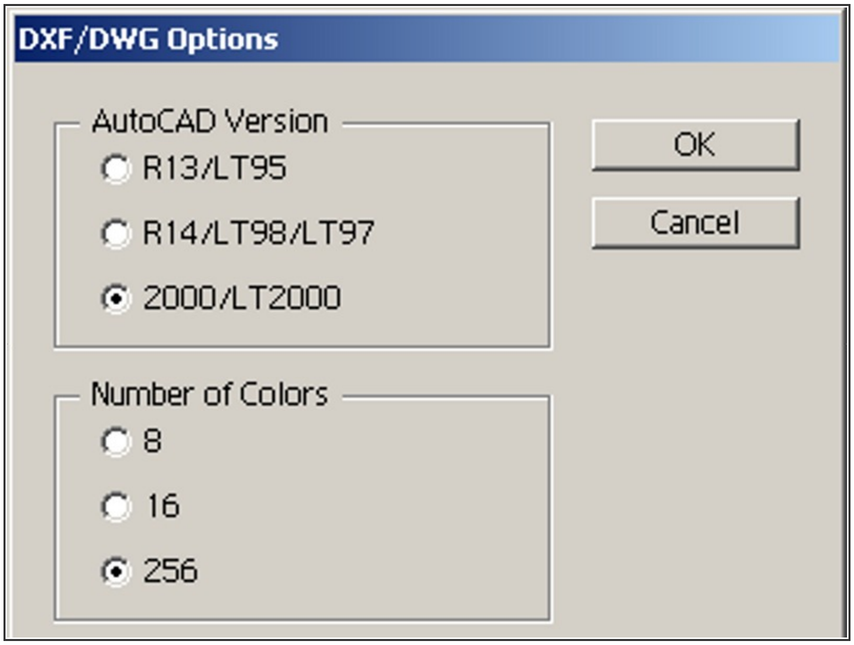
- The DXF importer in OpenSCAD is based on AutoCAD's DXF revision 12 (R12) file specification, which does not include some of the more advanced drawing elements included in revisions 13 and later. Most notably, these are the POLYLINE and SPLINE elements.
- It is doubtlessly not an accident that my version of Illustrator (CS3) only exports DXFs revision 13 and later, but these seem to be backwards-compatible with R12 as long as you avoid using certain drawing tools, notably compound paths. Put simply, each line in the drawing has to be a separate line segment with its endpoints unjoined. The box and ellipse tools seem to work, also.
- Make sure that snapping is turned on, and that the beginning of each line segment is snapped exactly to the endpoint of the preceding line segment. Otherwise your part profile will have openings and will not render correctly. If you screw up and include an unsupported drawing element or create a non-closed part profile, you'll get a warning later when OpenSCAD tries to render the model.

Step 4 — Delete grid layer


Adobe Illustrator
File Edit Object Type Select Filter Effect View Window Help
No Selection
rabbit-2 @ 600% (CMYK/Preview)
Layers
Layer 2
1 Layer

- When you've got the profile drawn to your satisfaction, save your drawing in its native format (.AI in the case of Illustrator).
- Then select the bottom layer containing the grid or other drawing guides and delete it, leaving only your part profile in the drawing's single remaining layer.

Step 5 — Export as DXF



DXF/DWG Options

AutoCAD Version

○ R13/LT95

○ R14/LT98/LT97

● 2000/LT2000

Number of Colors

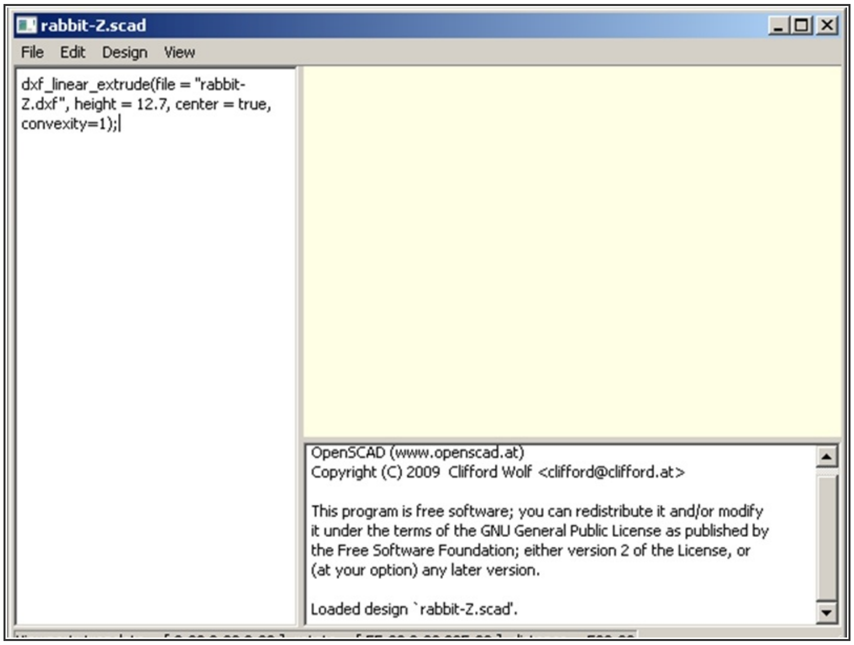
○ 8

○ 16

● 256

- If you have the option to export in DXF R12, use it.
- In Illustrator CS3, just use the default DXF export options.

Step 6 — Import DXF file into OpenSCAD



```
dxf_linear_extrude(file = "rabbit-Z.dxf", height = 12.7, center = true, convexity=1);
```

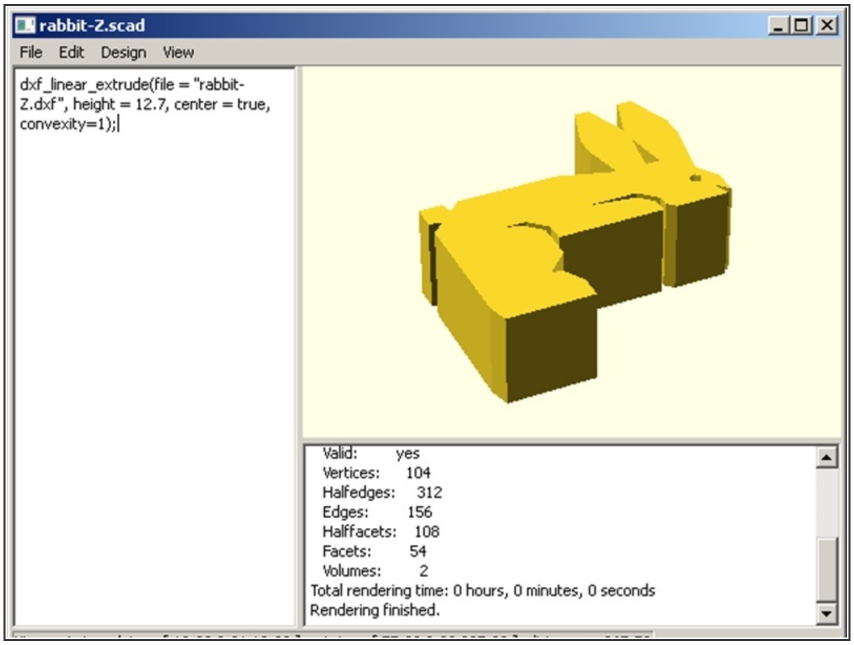
```
OpenSCAD (www.openscad.at)
Copyright (C) 2009  Clifford Wolf <clifford@clifford.at>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

Loaded design `rabbit-Z.scad'.
```

- Launch OpenSCAD, then enter (or copy-paste) the following code into the new document:
  - `dxf_linear_extrude(file = "rabbit-Z.dxf", height = 12.7, center = true, convexity=1);`
- "file" is the name of the DXF file you want to import, "height" is the height in millimeters you want to extrude the profile (12.7 mm = 0.5"), "center" is a Boolean variable that tells OpenSCAD whether to center the rendered model or not (I've never really messed with it), and "convexity" is a scalar representing the fineness with which to render curved surfaces. Given that my part profiles contain almost no curved surfaces, I was happy with the minimum convexity of 1. If your parts are curvier than mine you may want to raise it.
- Save the OpenSCAD script as a .SCAD file with an appropriate name.

Step 7 — Compile and render model

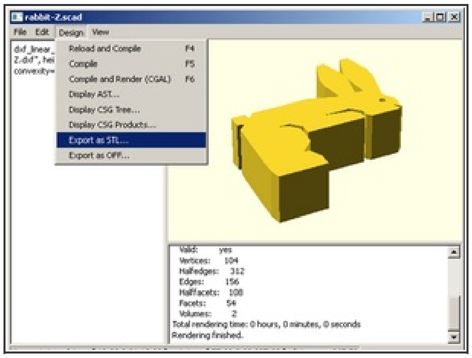


```
dxf_linear_extrude(file = "rabbit-Z.dxf", height = 12.7, center = true, convexity=1);
```

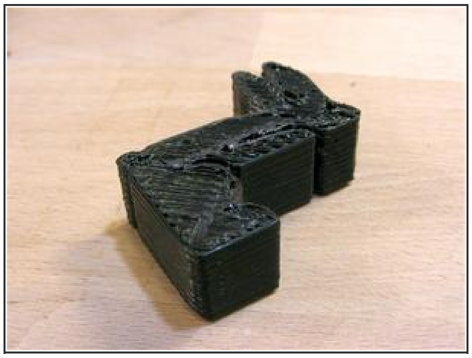
```
  Valid:        yes
  Vertices:     104
  Halfedges:    312
  Edges:        156
  Halffacets:   108
  Facets:       54
  Volumes:      2
Total rendering time: 0 hours, 0 minutes, 0 seconds
Rendering finished.
```

- From the drop-down menus select "Design > Compile and render (CGAL)" or just press F6.
- If you've done everything right, your extruded part should appear in the render window in the upper-right corner of the workspace, as shown.
- You can rotate it around by left-clicking and dragging the mouse, pan the camera by right-clicking and dragging, and zoom in and out with the scroll wheel.
- If the DXF file contains open paths or unsupported drawing elements, you will see corresponding error messages in the log window at the lower right. You may have to scroll up to see them. For whatever reason, even though you have produced only a single part, OpenSCAD will probably report that you have 2 volumes in your rendered model. This may or may not be a bug; in any case, it doesn't seem to hurt anything so don't worry about it.

Step 8 — Export model as STL file



```
Valid:        yes
Vertices:     104
Halfedges:    312
Edges:        156
Halffacets:   108
Facets:       54
Volumes:      2
Total rendering time: 0 hours, 0 minutes, 0 seconds
Rendering finished.
```



- From the drop-down menus, select "Design > Export as STL...", choose an appropriate filename, and click "Save."
- OpenSCAD should report "STL export finished" in the log window. And that's it! Your 3D model should be ready for printing or import into a compatible 3D WYSIWYG modeler like Blender.

OpenSCAD is still very much a work-in-progress. Although it crashed on me a couple of times when I tried to render DXF files with unsupported drawing elements, and I had a helluva time figuring out that the units of the Z-extrusion height were millimeters, generally I was very pleased with its performance and ease of use. If you have problems that you can't solve by experimentation, or that are clearly bugs, you can report them and/or get help by subscribing to the OpenSCAD mailing list here. There is also a wiki here. OpenSCAD is already very popular in the MakerBot community, and their Google group can also be a helpful source of information.

This document was last generated on 2012-10-31 11:58:15 PM.